Personal Recommendation Algorithm Combining Trust and Similarity based on Neural Network

La-Sheng Yu [1], Lei Zhang [2]
School of Information Science and Engineering,
Central South University,
Changsha, Hunan Province, China

***Abstract*— Recommendations based on friend trust relationship in social network have been extensively researched in recent years. However, the majority of the trust-based models merely take credibility or similarity into account, ignoring the influences of divergent factors on the recommendation results. This paper proposes a random walking recommendation algorithm based on conditional restricted Boltzmann machine in trust network, namely CRBM_PrTW. The algorithm fills the missing data in training data by utilizing the conditional restricted Boltzmann machine to improve the accuracy of similarity calculation, which effectively solves data sparsity problem. And on this basis, the comprehensive weight of the credibility, similarity and the trust factor utilized as trust level has been implemented to random walking algorithm. Therefore we present the random walking method based on credibility in trust network, which enhances the accuracy of recommendation system. The experimental result on the Epinions dataset demonstrates that our method can provide better recommendation result in terms of evaluation metrics when compared with the existing methods.**

Keywords*— *Conditional Restricted Boltzmann machine; Trust network; Random walking; Data sparsity

I. INTRODUCTION

With the rapid development of the Internet, the large amount of data generated by mobile terminals and Web services make it difficult for users to get the information they need. With innumerable online goods, users have to devote a lot of time to select what they favor. In order to meet the users, recommendation systems come into being, the major E-commerce sites employing the recommendation system to furnish users with products that may interest them, such as Taobao, Jingdong and Amazon, etc. They have offered personalized recommendation service.

Collaborative filtering (CF) is the most extensively applied and successful recommendation algorithm [1,2,3]. Mainstream collaborative filtering algorithms are divided into two categories: user-based CF [4] and item-based CF [5]. The user-based collaborative filtering calculates the similarity between users based on the their scoring matrix and finds a collection of users sharing similar interests. Then it recommends to the target users the identified items that are preferred by other users but not found by target users in the collection. But if only few users rate on the item, the rating information is extremely sparse, then it is hard to find a similar user set. Item-based collaborative filtering predicts the rating of target items in conformity with the user's rating data on similar items. However, the recommendation coverage is low as the user's interests change constantly.

The Restricted Boltzmann Machine (RBM) can be regarded as an undigraph model [6]. In recent years, the RBM as the basic component module has achieved great success in deep learning, meanwhile the RBM model has been demonstrated to help address the cold start problem in recommendation system. Salakhutdinov et al. [7] initially applied the RBM model to the recommendation system and proposed the Conditional Restricted Boltzmann Machine (CRBM) model. The CRBM model makes full use of the rated/unrated information to mitigate the negative impacts of the sparse data on recommendation results. Liu et al. [8] extended RBM model by incorporating content-based features such as user demographic information, items categorization and other features. The experimental results show that Content-boosted Restricted Boltzmann Machine (CB-RBM) performs better than a pure RBM model and other content-boosted collaborative filtering methods.

In recent years, with the advancement of social network, the usage of user relationship in social network has become a research focus in recommendation field. People are more inclined to receive recommendations from their closest friends in social networks. Golbeck [9] proposed a TidalTrust model to use modified breadth first search strategy in the Trust network to infer the trust value between the source users and others, but it ignores the impacts of the part of raters far away from the source users on the recommendation results. Massa et al. [10,11] presented a ModelTrust model that is similar to TidalTrust. But ModelTrust considers all users in the pre-difined range and calculates the trust among users as the weight value. The biggest challenge for trust based recommendation is to select the distance to explore in the trust network. To solve this problem, Jamali and Ester [12] put forward a random walk algorithm in the trust network, namely the TrustWalker algorithm. However, the majority of the trust models only consider the single factor of trust or similarity but the effects of various factors on recommendation result.

In order to tackle the problems mentioned above, this article, combining both CRBM and TrustWalker, presents a CRBM_PrTW algorithm that can adapt to the sparse data and random walk in the trust network. The CRBM_PrTW algorithm completes the training data through the CRBM model and calculates the user similarities and the comprehensive weights of the trust, similarity and credibility. By applying random walking in trust network, the evaluated target items and the items similar to the target items are considered, which enhance the accuracy and coverage of recommendation results.

II. RELATED WORKS

Data sparsity means the rating matrix gradually becomes sparse with the growth of users and items, resulting in inaccurate calculation of item similarity. In order to handle the problem, some improved methods adopted the way of pre-filling rating matrix. The singular value decomposition (SVD) model is proposed to reduce the dimension of the rating matrix and utilizes the dense rating matrix to recommend [13]. The paper [14] points out that the application of Principal Component Analysis (PCA) can alleviate the data sparsity problem. A method based on K-means clustering is presented [15,16], which gathers users with similar interest into the same cluster and uses the average score of them to predict the missing score in the rating matrix, and to some extent solves the sparse data problem. The paper [17] applies the auto-encoders to study the low dimensional features to forecast the rating of the item. Koren et al. [18] proposed a matrix decomposition model, which decomposes a sparse matrix into two low order sub-matrices. Through continuous iterative training, the product of two matrices is getting closer to the real rating matrix.

The TidalTrust uses the improved breadth first search method considering the shortest path in the trust network to obtain the trust value between users by weighting the trust value [9]. However, the model neglects the influence of the users who are far away from the source users on the recommendation results. The ModelTrust algorithm [10,11] is similar to the TidalTrust besides that a maximum path length is preset. At the basis of the data in real network, Ziegler et al. [19] analyze the similarity and the trust relationship among users and demonstrate that there is a positive correlation between them. Jamali and Ester [12] propose a random walk model in the trust network. In the course of walking, both the target users and the user's rating on the items that are similar to the target item are considered, which solves data sparsity and cold start problem. But it is of equal probability to select the next user in random walk. However, the reality is that if a user and the source user are of credibility and share greater similarity, it is more likely to be selected in the walking. The users with closer social relationships to others are much worth to be believed [20] and are much powerful in affecting others [21]. The paper [22] considers divergent factors of recommendation resources and proposes the electronic commerce recommendation system incorporated with trust and social relations.

The traditional solutions mentioned above all have various shortcomings and can not effectively solve calculation accuracy of user similarity under extremely sparse user item rating.This paper utilizes the CRBM to extract the features and predicts the missing data in the rating matrix. And it builds a robust trust network to improve the recommendation accuracy by applying the modified random walking algorithm in the trust network.

III. ALGORITHM DESCRIPTION

A. *Similarity calculation*

The correlation between users and items is the key of collaborative filtering algorithm, therefore the usage of an appropriate method to measure similarity is paramount to obtain accurate recommendations. Currently there are several commonly used similarity calculation formulas: Jaccard similarity coefficient, Pearson correlation coefficient [12], cosine similarity and modified cosine similarity [5]. Pearson similarity between item i and item j:

$$corr(i,j)=\frac{\sum_{u\in UC_{i,j}}(r_{u,i}-\bar{r}_u)(r_{u,j}-\bar{r}_u)}{\sqrt{\sum_{u\in UC_{i,j}}(r_{u,i}-\bar{r}_u)^2}\sqrt{\sum_{u\in U}(r_{u,i}-\bar{r}_u)^2}} \quad (1)$$

Here, $UC_{i,j}$ represents a set of common users who have rated both item i and item j, $\bar{r}_u$ is the average of ratings made by user u.

Pearson similarity only considers the rating divergences among items but the influence of the number of the common users. For instance, if $corr(i,j)=corr(i,l)$, but $|UC_{i,j}|>|UC_{i,l}|$, then there are more common users between item i and item j than item i and item l, the correlation between item i and item j is stronger and $sim(i,j)>sim(i,l)$. Therefore, we consider $UC_{i,j}$ in the similarity measure as follows:

$$sim(i,j)=\frac{1}{1+e^{-\frac{|UC_{i,j}|}{2}}}\times corr(i,j) \quad (2)$$

Cosine similarity does not take into account the user rating scale. For instance, in the rating range [1-5], user u rates 3 points to show like, and user v, over 4 points. The user's average score is subtracted in cosine similarity to tackle the problem. The similarity between user u and user v is:

$$sim(u,v)=\frac{\sum_{c\in I_{u,v}}(R_{u,c}-\bar{R}_u)(R_{v,c}-\bar{R}_v)}{\sqrt{\sum_{c\in I_u}(R_{u,c}-\bar{R}_u)^2}\sqrt{\sum_{c\in I_v}(R_{v,c}-\bar{R}_v)^2}} \quad (3)$$

Here, $R_{u,c}$ denotes user u 's rating on item c, $\bar{R}_u$ and $\bar{R}_v$ is the average ratings on the item rated by user u and user v respectively.

B. *Conditional Restricted Boltzmann Machine*

The traditional collaborative filtering based on RBM failed to consider a significant factor: there are some items users have rated or watched while we don't have ratings. This implicit information also provides additional insight into a user's preferences. For example, if a user has evaluated "Rocky 5", we can conjecture that the user likes the type of movies.

The Conditional RBM model takes the information into consideration. Define $r\in\{0,1\}^M$ is a binary vector of Length M (the total number of movies) , indicating all the movies the user rated (the ratings are unknown). The idea is to define a joint distribution on r that exceeds the condition (V,h). In the conditional RBM model, the vector r affects the state of the hidden units in Figure 1.

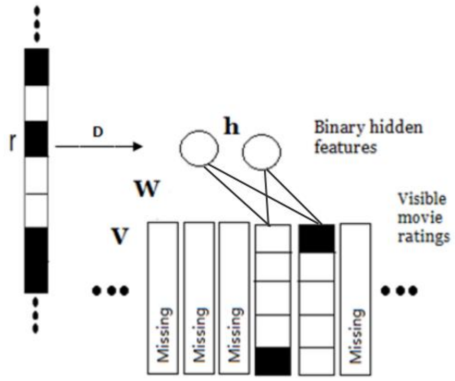

Fig. 1. Conditional RBM. The binary vector r, indicating rated/unrated movies, affects binary states of the hidden units.

The model can be considered as a undirected graph model, V is the visible layer representing the data; h is the hidden layer, the feature extractor; W is the connection weight matrix between the visible and the hidden layers, and D is the connection weight matrix between the r layer and the hidden layer; While C is the bias of the visible unit, b is the bias of the hidden unit.

$$p(v_i^k = 1 | h) = \frac{\exp(c_i^k + \sum_{j=1}^{F} h_j w_{ij}^k)}{\sum_{l=1}^{K} \exp(c_i^l + \sum_{j=1} Fh)} \qquad (4)$$

$$p(h_j = 1 | V, r) = \sigma(b_j + \sum_{i=1}^{m} \sum_{k=1}^{K} v_i^k W_{ij}^k + \sum_{i=1}^{M} r_i D_{ij}) \qquad (5)$$

(5)

Welling and Hinton proposed a RBM fast learning algorithm, namely the Contrastive Divergence (CD) algorithm [23], the algorithm is also implemented to update parameter in CRBM model. On the basis of other research achievements [2,25], we train CRBM with hidden units F=100 and features C=40 in the following experiments.

C. Recommendation Method of Random Walking in Trust Network Based on Trust Factor

1) Problem Definition

In recommendation systems we have a set of users $U = \{u_1, .. u_N\}$ and a set of items $I = \{i_1, .. i_M\}$. Each user u rates a set of items $RI_u = \{i_{u1}, .. i_{uk}\}$. The rating of user u on item i is denoted by $r_{u,i}$. $r_{u,i}$ can be any real number, but often ratings are integers in the range [1, 5]. If user u trusts the user v, then $t_{u,v}$ denotes the value of the trust as a real number in [0, 1]. Zero means no trust, and one means full trust. In addition, we define $TU_u = \{v \in U \mid t_{u,v} = 1\}$ as the user set directly trusted by user u. TN_v denotes the total number of users of user v. The trust network can be defined as a graph $G = \langle U, TU \rangle$, and the directed edges in the graph is illustrated as $TU = \{(u,v) \mid u \in U, v \in TU_u\}$.

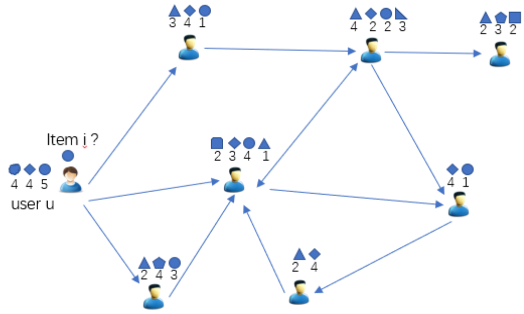

Fig. 2. Illustration of a social network including ratings expressed by users. The ratings are shown below the item icon beside the user.

As shown in this figure, we have a social network of users. Each user has rated some items. The source user u wants a prediction on the target item i. Usually only a small number of users have rated the target item i, other users have ratings on other items which may include items already rated by user u or items similar to the target item i.

2) A Single Random Walk

In order to predicting user u_0's rating on item i, we will perform our random walk in trust network with u_0 as the source user. When walking to a certain user u, if u has rated target item i, then we stop the random walk and return to the item $r_{u,i}$ as the result of random walk. If u does not have a rating on the item i, then we have two options:

With probability $\phi_{u,i,k}$, we don' t continue the random walk. We randomly select one of the item j that is similar to the item i rated by u and we return $r_{u,j}$ as the result of this random walk. The probability of selecting j from items rated by user u is $P(Y_{u,i} = j) = \frac{sim(i,j)}{\sum_{l \in RI_u} sim(i,l)}$, $Y_{u,i}$ indicates a random variable for item j among items rated by u while looking for an item similar to target item i. $sim(i,j)$ denotes the similarity between item i and item j, and $\phi_{u,i,k}$ represents the probability to stop walking. Details will be discussed later.

With probability $1\text{-}\phi_{u,i,k}$, we continue our random walk from the current user u to another user v who is one of user u's directly trusted neighbors ($v \in TU_u$). Therefore before selecting the next user, we need to calculate the trust weight $TW(u,v)$ for all the users to be selected:

$$TW(u,v) = (\alpha * t(u,v) + (1-\alpha) * sim(u,v)) * \frac{1}{1+e^{-TN_v}} \qquad (6)$$

Here, the trust value between users is illustrated by $t(u,v)$. The Trust network is a two-valued network on Epinions dataset, one means trust and zero means distrust. $sim(u,v)$ is the similarity between user u and the user v. The formula introduces a trust factor TN_v, representing the total number of users who trust v. In the trust network, mutual trust users do not always share high similarity of interest. As the number of users who trust user v grows, the recommendation from user v will be more credible. Therefore, we propose the improved probability formula through combining together the trust $t(u,v)$, similarity $sim(u,v)$, trust factor TN_v and sigmoid function. We define a random variable S_u for a user v to select from TU_u.

$$P(S_u = v) = \frac{TW(u,v)}{\sum_{w \in TU_u} TW(u,w)} \qquad (7)$$

3) Termination of a Single Random Walk

We have a stop-walk probability $\phi_{u,i,k}$ for staying at each user u, and k is walking depth. Furthermore, ratings on target item i from users far away from source user u_0 are noisy, but ratings expressed by trusted users nearby in the network are more reliable. Another factor that influences $\phi_{u,i,k}$ is the maximum similarity $\max(sim(i,j))$ between item i and item j rated by user u. So the deeper we go into the network, the probability of continuing our random walk should decrease and so $\phi_{u,i,k}$ should increase. The formula is as follows:

$$\phi_{u,i,k} = \max_{i \in RI_u}(sim(i,j)) \times \frac{1}{1+e^{-\frac{k}{2}}} \qquad (8)$$

Each random walk has three alternatives to stop:

(1) Arriving at a node which has expressed a rating on the target item i.

(2) When moving to the node u without rating on the target item i, we stop the walk at probability $\phi_{u,i,k}$ and return the rating for a similar item .

(3) If there is not restriction for walk depth, the walk may continue forever. To avoid such a case in our implementation,this paper sets the maxi-depth as 6 steps based on the theory of "six degree segmentation" [23]. When the walk depth comes to 6 steps, the walking is terminated.

4) Termination Condition of the Overall Walk

In order to get an accurate prediction rate $r_{u_0,i}$, we need to select an appropriate overall walk depth. The formula is utilized to calculate the variance of in the results of all the walks:

$$\sigma^2 = \frac{\sum_{i=1}^{T}(r_i - \bar{r})^2}{T} \qquad (9)$$

Here, r_i denotes the result of the i^{th} random walk, and T indicates the number of random walks we perform to compute the prediction. $\bar{r}$ represents the average of the ratings returned by random walks. We define σ_i^2 as the variance of the i random walk and the rating finite range of the item is $[1,5]$, it can be demonstrated that σ^2 tends to a constant value. So we can terminate random walk when $|\sigma_{i+1} - \sigma_i| \le \zeta$.

5) Recommendation Results Generation

At the termination of each random walk, we will choose a user v ' s rating $r_{v,j}$ on item j as the result of a single random walk, of which item j may be the same as target item i or not. When the overall walk is done, we can get the forecast rate of the source user u_0 on target item i:

$$\hat{r}_{u_0,i} = \sum_{k=1}^{K} W_k r_k \qquad (10)$$

Here, W_k is the weight of the k^{th} random walk, r_k denotes the return result of the k^{th} random walk. Formula 11 is the probability of walking from the initial node u_0 to the current node u_v and returning to item j:

$$\Pr_K = \phi_{v,i,K} P(Y_{v,i} = j) \prod_{\{(u_s,u_e) \in TU, k=1\}}^{K} (1 - \phi_{u_s,i,k}) P(S_{u_s} = u_e) (11)$$

Here, u_s and u_e are directly trusted friends, and $u_s = u_0$ initially. The weight of each random walk is:

$$W_K = (1/\log(\Pr_K)) / \sum_{k=1}^{K} \log(\Pr_k) \qquad (12)$$

D. *Improved Random Walk Algorithm Based on Conditional Restricted Boltzmann Machine*

Trust based recommendation algorithm has been extensively studied in recent years in terms of data sparse and cold start. However, There are still some problems.

Similarity calculation is commonly implemented in a variety of recommendation algorithms, But the results lack accuracy when the data is sparse.

2) Mutually trusted friends may hold diverse interests. Trust users are not always similar, and vice versa. Traditional trust-based recommendation algorithms usually consider only trust or similarity. For example, in the TrustWalker model, the target user u 's prediction score for the target item i is calculated using only trust as a weighting coefficient, ignoring the effects of the user's preference similarity on recommendations.

In order to address the above shortcomings, this paper proposes a Random Walk Recommendation Algorithm Based on Conditional Restricted Boltzmann Machine in the trust network (CBRM_PrTW). The algorithm applies the CRBM to predict the the missing value in the user item rating matrix and the trust matrix, which improves the accuracy of the user similarity calculation and constructs a more comprehensive social network. In the process of random walk, we take into account the trust, user similarity and confidence factor to calculate trust weigh. And the recommendation is obtained with the upgraded random walk algorithm.

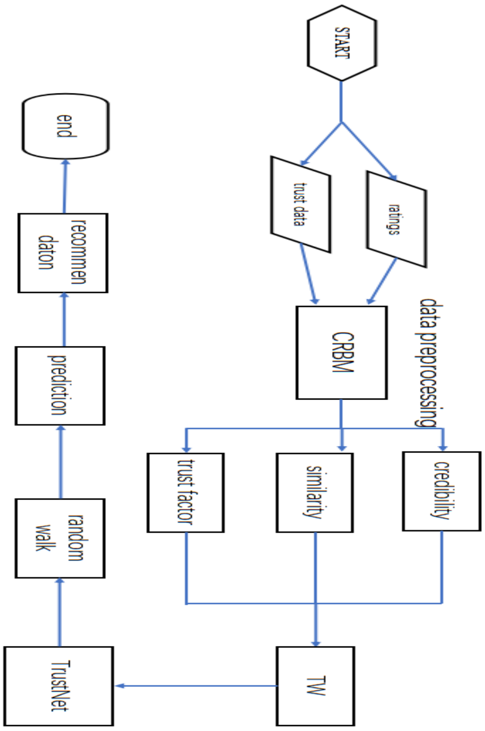

Fig. 3. CRBM_PrTW algorithm flow chart.

Algorithm 1. The description of the CRBM_PrTW algorithm.

Input: training dataset, rated/unrated data

Output: prediction rate

```
    // TrustNet represent Trust Network
Step1: Training a CRBM model
Step2: For each $u \in U$ do:
//Use rating vector $RI_u$ as an input to the constructed CRBM model;
CRBM← $RI_u$
Select the items to be rated to predict the vacant values
Fill the vacancy of $RI_u$ with the prediction value
End for
Step3: Call the CreateTrustNet function to build TrustNet
Step4: foreach $r_{u_0,i}$ which need to be predicted random walk in TrustNet:
Step5: if $|\sigma_{i+1} - \sigma_i| \le \zeta$ or K>6:   //K denotes walking steps
Stop random walk
The source user $u_0$'s prediction rating for the target item $i$:
$\widehat{r}_{u_0,i} = \sum_{k=1}^{K} W_k r_k$
Step6: if ($u$ has rated $i$):
        Stop random walk
        return $r_{u,i}$
Else:
If (random(0,1)<$\phi_{u,i,k}$ || K=6):
        stay at user u
with probability $P(Y_{u,i} = j) = \frac{sim(i,j)}{\sum_{l \in RI_u} sim(i,l)}$,
select item j
        Return $r_{u,j}$
else:
   //v is one of u’ s direct trusted friends
   According to formula (7), walk to user v, K++
   u←v   //Assign v to u
   goto Step5
end for
```

Algorithm 2. The description of the CreateDirectTrustNet algorithm.

Input: CRBM model, training dataset, rated/unrated data

Output: TrustNet

```
Step1: for all Friendship ($u_1, u_2$) in social network do:
calculate the similarity between user $u_1$ and user $u_2$:
$sim(i,j) = \frac{1}{1+e^{-\frac{|UC_{i,j}|}{2}}} \times corr(i,j)$
calculate trust weight:
$TW(u,v) = (\partial * t(u,v) + (1-\partial) * sim(u,v)) * \frac{1}{1+e^{-TN_v}}$
add the trust weight to the trust network
end for
```

Rating data and trust data in Algorithm 1 are used as the input of CRBM model to predict the missing value, and it solves the sparsity of training set. We calculate the trust weight by taking a comprehensive of trust, similarity and credibility to build a more sound trust network. That is to say, mutually-trusted users do not always hold similar interests and users sharing low similarity may be congenial to each other’ s tastes. What’s more, we tend to select users with higher credibility in random walking. After K random walks, we regard the weight value of results returned by each walk as the final prediction value.

E. Experimental Results and Analysis

1) Experimental data

This paper selects the Epinions dataset obtained from the Epinions website to test the validity of algorithm. There are in total 49,290 users, 139,738 items and 664,824 rating records in the data set. What’ s more, the rating sparsity amounts to 99.99% and the average rating records are 13.3 times per user. The total number of trust record contains 487,181, with 9.9 directly-trusted friends per user. Therefore, the Epinions dataset is extremely sparse and can be used to evaluate the recommendation results of the model under the condition of utmost data sparsity.

2) Evaluation metrics

In this paper, Root Mean Square Error (RMSE) is one of the evaluation metrics and the formula is :

$$RMSE = \frac{\sqrt{\sum_{u,i\in Test}\left(r_{u,i} - \widehat{r}_{u,i}\right)}}{|Test|} \qquad (13)$$

Here, $r_{u,i}$ denotes the user u 's real rating on item i , $\widehat{r}_{u,i}$ is the prediction rating and $|Test|$ means the number of items rated by target user u . The smaller the value of the $RMSE$, the higher the accuracy of the recommendation results.

In the case of extreme data sparsity, some models recommend only a small number of movies to users. Therefore, this chapter selects Coverage as the second metric for evaluation metrics. M represents all the movies in the test set, and $R(u)$ denotes the movies recommended to users. The coverage calculation formula is:

$$coverage = \frac{|R(u)|}{|M|} \qquad (14)$$

To combine RMSE and coverage into a single evaluation metric, we calculate F-Measure [12]. So we need to transformed RMSE into a precision metric in the range [0,1]. The precision and F-Measure are calculated as follows:

$$\Pr ecision = 1 - \frac{RMSE}{4} \qquad (15)$$

$$F = \frac{2\times precision \times converage}{precision + converage} \qquad (16)$$

3) Coefficient selection of the trust weight

In this paper, the coefficient α is of significance to calculate trust weight TW , the parameter affects the recommendation accuracy. So we need to analyze the relationship between parameter value and recommendation accuracy. In the CRBM_PrTW, the coefficient α range is 0.1 to 1 in the experiment. The Figure 4 describes the variation of the RMSE when the coefficient α takes different values.

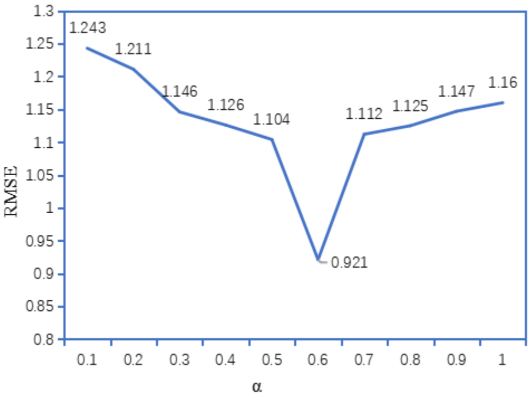

Fig. 4. Relationship between Trust Weight Coefficient α and RMSE.

As it is shown in the Figure 4, when the α ⩽0.6, the RMSE gradually decreases with the increase of α ; when α >0.6, the RMSE also increases with the growth of α ; when α equals 0.6, the RMSE is the minimum value. In the following experiment, we select parameter α =0.6 as the contrast experiment.

4) Experimental results

Considering the rigor of the experiment, we randomly divide the data set into 75% training set and 25% test set, and select the user-based CF, Item-based CF, TrustWalker, ModelTrust and TidalTrust algorithm mentioned above as contrast experiments.

TABLE I. EXPERIMENTAL RESULTS OF DIFFERENT RECOMMENDATION ALGORITHMS.

Method	RMSE	Converge(%)	F-Measure
User-based CF	1.252	63.24	0.283
Item-based CF	1.323	60.23	0.341
TrustWalker	1.124	86.12	0.742
ModelTrust	1.167	82.22	0.627
TidalTrust	1.135	79.61	0.643
CRBM_PrTW	0.921	90.15	0.852

As we can see from table 2, the CRBM_PrTW proposed in this paper is better than all other the algorithms in terms of evaluation metrics. The performance of traditional recommendation algorithm based on collaborative filtering is the worst. When the training set is extremely sparse, users merely rate on a small number of items and only a few items are scored by various users. It is difficult to calculate item similarity or user similarity, which reduces the recommendation quality. Trust network is utilized in the recommendation algorithm, as the items are recommended by its own trust users, the recommendation is significantly advanced in quality compared with the traditional collaborative filtering algorithm. The experiment demonstrates that the coverage and RMSE of TrustWalker are higher than that of ModelTrust and TidalTrust. That is due to the TrustWalker algorithm takes into account not only the trust user's rating of the target item, but also the rating of items similar to the target item. Our method utilizes CRBM to solve data sparse, which greatly improves the accuracy of similarity calculation. Meanwhile, trust value, user similarity and trust factor are considered to calculate trust weight that is applied in random walking algorithm, which enhances the recommended quality.

CONCLUSION

This paper proposed a random walk based on CRBM. The CRBM model effectively tackles data sparse and increases the accuracy of similarity calculation. In addition, this paper takes into consideration of credibility, user similarity and the trust factor. Therefore, the next selected users are trust users who share similar tastes in the random walk algorithm. Meanwhile, the trust factor makes the recommendation result more reliable. The experimental results show that the proposed CRBM_PrTW algorithm is better than other recommended methods.

There are several aspects that need to be considered in the future: firstly the trust in this article is binary trust, while in daily life, trust between people can be divided into different levels, such as trust, general trust, very trust and etc. We intend to calculate the type of real trust through trust propagation. Secondly, users share divergent interests in different types of movies, so we introduce the category factor in the method.

ACKNOWLEDGMENT

This research was supported by National Natural Science Foundation of China (Grant No. Z201610110620003). We are very grateful to Hengjuan Wang for her proof reading this paper.

REFERENCES

[1] Goldberg, David, et al. "Using collaborative filtering to weave an information tapestry." Communications of the ACM 35.12 (1992): 61-70.

[2] Kim, Kyoung-jae, and Hyunchul Ahn. "Recommender systems using cluster-indexing collaborative filtering and social data analytics." International Journal of Production Research (2017): 1-13.

[3] Cao, Cen, Q. Ni, and Y. Zhai. "An Improved Collaborative Filtering Recommendation Algorithm Based on Community Detection in Social Networks." Conference on Genetic and Evolutionary Computation ACM, 2015:1-8.

[4] Ma, M. M, and S. P. Wang. "Research of User-Based Collaborative Filtering Recommendation Algorithm Based on Hadoop." International Conference on Computer Information Systems and Industrial Applications 2015.

[5] [5] Li, Chenyang, and K. He. "CBMR: An optimized MapReduce for item-based collaborative filtering recommendation algorithm with empirical analysis." Concurrency & Computation Practice & Experience 29.10(2017):-.

[6] Fischer, Asja, and C. Igel. "An Introduction to Restricted Boltzmann Machines." 7441(2012):14-36.

[7] Salakhutdinov, Ruslan, A. Mnih, and G. Hinton. "Restricted Boltzmann machines for collaborative filtering." International Conference on Machine Learning ACM, 2007:791-798.

[8] Liu, Yongqi, et al. Content-Boosted Restricted Boltzmann Machine for Recommendation. Artificial Neural Networks and Machine Learning – ICANN 2014. Springer International Publishing, 2014:773-780.

[9] Golbeck, Jennifer Ann. Computing and applying trust in web-based social networks. University of Maryland at College Park, 2005.

[10] Massa, Paolo, and P. Avesani. "Trust-aware recommender systems." ACM Conference on Recommender Systems ACM, 2007:17-24.

[11] Massa, Paolo, and P. Avesani. Trust-Aware Collaborative Filtering for Recommender Systems. On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE. Springer Berlin Heidelberg, 2004:492-508.

[12] Jamali, Mohsen, and Martin Ester. "Trustwalker: a random walk model for combining trust-based and item-based recommendation." Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009.

[13] Zhang, Sheng, et al. "Using singular value decomposition approximation for collaborative filtering." E-Commerce Technology, 2005. CEC 2005. Seventh IEEE International Conference on. IEEE, 2005.

[14] Goldberg, Ken, et al. "Eigentaste: A constant time collaborative filtering algorithm." Information Retrieval 4.2 (2001): 133-151.

[15] You, Haipeng, et al. "An Improved Collaborative Filtering Recommendation Algorithm Combining Item Clustering and Slope One Scheme." Proceedings of the International MultiConference of Engineers and Computer Scientists. Vol. 1. 2015.

[16] Beutel, Alex, Amr Ahmed, and Alexander J. Smola. "Accams: Additive co-clustering to approximate matrices succinctly." Proceedings of the 24th international conference on world wide web. International World Wide Web Conferences Steering Committee, 2015.

[17] Yi, Baolin, et al. "Expanded autoencoder recommendation framework and its application in movie recommendation." Software, Knowledge, Information Management & Applications (SKIMA), 2016 10th International Conference on. IEEE, 2016.

[18] Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." Computer 42.8 (2009).

[19] Ziegler, Cai-Nicolas, and Georg Lausen. "Analyzing correlation between trust and user similarity in online communities." ITrust. Vol. 2995. 2004.

[20] Chau, Michael, and Jennifer Xu. "Mining communities and their relationships in blogs: A study of online hate groups." International Journal of Human-Computer Studies 65.1 (2007): 57-70.

[21] Lee, Matthew KO, et al. "Consumer's decision to shop online: The moderating role of positive informational social influence." Information & Management 48.6 (2011): 185-191.

[22] Li, Yung-Ming, Chun-Te Wu, and Cheng-Yang Lai. "A social recommender mechanism for e-commerce: Combining similarity, trust, and relationship." Decision Support Systems 55.3 (2013): 740-752.

[23] Welling, Max, and Geoffrey E. Hinton. "A new learning algorithm for mean field Boltzmann machines." International Conference on Artificial Neural Networks. Springer, Berlin, Heidelberg, 2002.

[24] Travers, Jeffrey, and Stanley Milgram. "The small world problem." Phychology Today 1 (1967): 61-67.

[25] Georgiev, Kostadin, and Preslav Nakov. "A non-iid framework for collaborative filtering with restricted boltzmann machines." International Conference on Machine Learning. 2013.